Programming Engineering

Course 1 – 20 February 2017

Adrian Iftene
adiftene@info.uaic.ro

Content

- Context
- Motivation
- Famous errors
- Statistics
- Definitions
- Stages of development programs
- Developing models
  - Waterfall
  - Spiral

Context

- **Large applications** (millions of lines of code) written over several months, years ...
- **Working teams**: Project managers, analysts, architects, programmers, testers, support engineers (the order of 10, 100, 1,000 people ...)
- The solution to these problems is a solution written in an **object-oriented programming language**

Motivation 1

- More and more systems are controlled by software: traffic control (air, rail, road, water, etc.), banks, mobile telephony, Internet, supermarket, smart homes
- Economies of all states depend on software
- Software Engineering proposes theories, methodologies and tools for professional software development

Motivation 2

- **1946** Goldstine and von Neumann: "1000 guidelines are a reasonable upper limit to the complexity of issues which can be designed as done by computer"

- Ticket booking system for airline KLM contained, in **1992**, **two million lines of code** in assembly language

Motivation 3

- System V operating system version 4.0 (UNIX) was obtained by compiling the 3.7 million lines of code
- Programs written for the NASA space shuttle had around 40 million lines of code object
- To create the IBM OS360 operating system 5,000 man-years were required

Motivation – Famous errors 1

- Huge electricity bills for pensioners, recalculation of pensions

- IBM OS360 contained in each relaunching 1,000 mistakes. *Resignation*...

Motivation – Famous errors 2

- *Unlucky programmer at a bank:* Bank wanted to send a special letter to important customers by mail to notify them various services
- The programmer wrote a program that selected 2,000 customers and wrote a personalized letter
- In testing he used a pseudonym client **Rich Bastard**
- Sadly 2,000 customers have received a letter that began "**Dear Rich Bastard** ..."

Motivation – Famous errors 3

- Sacramento: dentist receives in the mailbox 16,000 forms for payment of fees in one week – *"It Was a computer problem"* an official said

- *"Failure to convert English measures to metric values was the root cause of the loss of the Mars Climate Orbiter…"*

Motivation – Famous errors 4

- Venus exploration vehicle loss. Ah, it was actually **.,.** in FOR! ...
- Missile warning system activated. **Attack** or not?
- Ariane 5 explodes. Fireworks Cost: **$ 500 million**

Motivation – Statistics 1

- These are not just "funny errors"

- Large projects may be the most complicated products made by anyone
- Let's look at the statistics
  - We consider that a software project is **successful** if it is done in a reasonable period of time and with a reasonable budget
  - A failure of a software product occurs when the product is not achieved or it can not be used

Motivation – Statistics 2

- **Successful studies: USA'82 – Gibson & Singer**
- **18 projects**
  - Success: 17%
  - Partially in use: 28%
  - Satisfactory 11%
  - Failure: 22%
  - Unrated: 11%
- **Reasons for failures**
  - Organizational problems
  - The new working methods and salary policies
  - Unforeseen changes in business

Motivation – Statistics 3

- Successful studies – ONNI'88 (Finland)

- From over 100 projects
  - Success: 33%
  - Problem: 42%
  - Failure: 25%

- Reasons for failures
  - Poor training of software engineers
  - Insufficient resources
  - Management issues

Motivation – Statistics 4

- **The Robbins-Gioia Survey (2001),** Alexandria – Virginia, made a study over the perception by enterprises of their implementation of an E.R.P. (Enterprise Resource Planning) package.
- 232 survey respondents (36 % had, or were in the process of, implementing an ERP system)
- **51 % viewed their ERP implementation as unsuccessful**,
- **46 % did not understand how to use the system**
- 56 % of survey respondents noted that their organization has a program management office (PMO) in place, and of these respondents, only 36 % felt their ERP implementation was unsuccessful

Motivation – Statistics 5

- **The Conference Board Survey (2001)**
- Of 117 companies that attempted ERP implementations
- **34 % were very "satisfied"**
- **58 % were "somewhat satisfied"**
- **8 % were unhappy** with what they got.
- **40 % of the projects failed** to achieve their business case within one year of going live

- The companies that did achieve benefits said that achievement took **six months longer** than expected.
- Implementation costs were found to average 25 % over budget

PE – Definition 1

- First definition (NATO,1968): *the establishment and use of sound engineering principles in order to economically obtain software that is reliable and works efficiently on real machines*

- IEEE Definition (*IEEE Standard Glossary of Software Engineering Technology, 1983): The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software*

PE – Definition 2

- Refers to two things:
  - "Software engineer" who replaced "programmer"
  - "Software Engineering" is used to describe "building of software systems which are so large or so complex that they are built by a team or by teams of engineers“ – *Fundamentals of Software Engineering* (Ghezzi, Jazayeri, and Mandrioli)

PE – Definition 3

- **FreeDictionary**: "The process of **manufacturing** software systems. A software system consists of executable computer code and the supporting documents needed to manufacture, use, and **maintain** the code."
- **Webopedia.com**: "The computer science discipline concerned with **developing large applications**. Software engineering covers not only the technical aspects of building software systems, but also **management issues**, such as **directing programming teams**, **scheduling**, and **budgeting**. "

PE – Definition 4

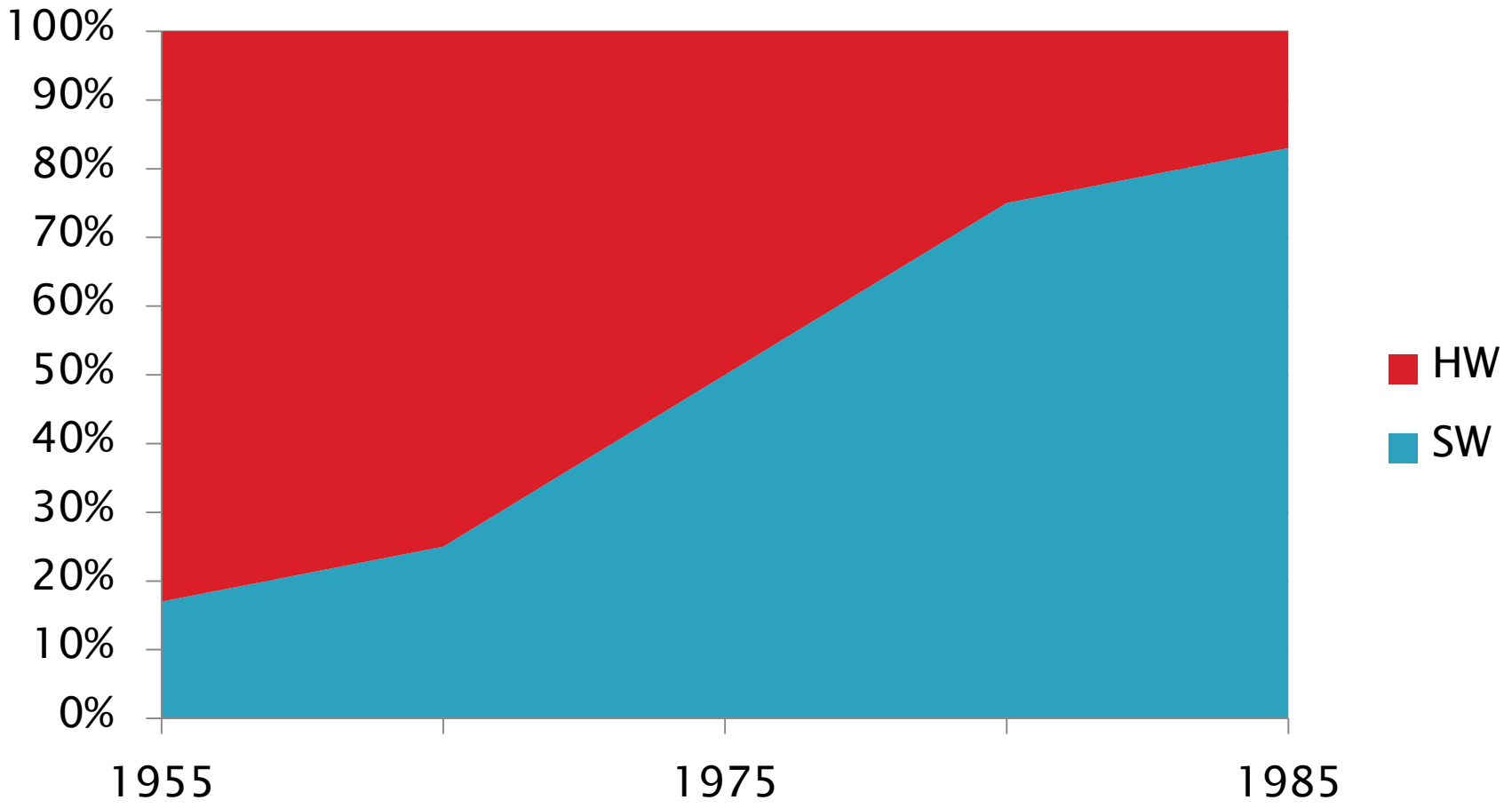
- **Wikipedia**: "**Software engineering** is the application of a systematic, disciplined, quantifiable approach to the **development**, operation, and **maintenance** of software, and the study of these approaches"

- **Answers.com**: "The **systematic** application of scientific and technological knowledge, through the medium of sound engineering principles, to the production of computer programs, and to the **requirements definition**, **functional specification**, **design description**, **program implementation**, and **test methods** that lead up to this code"

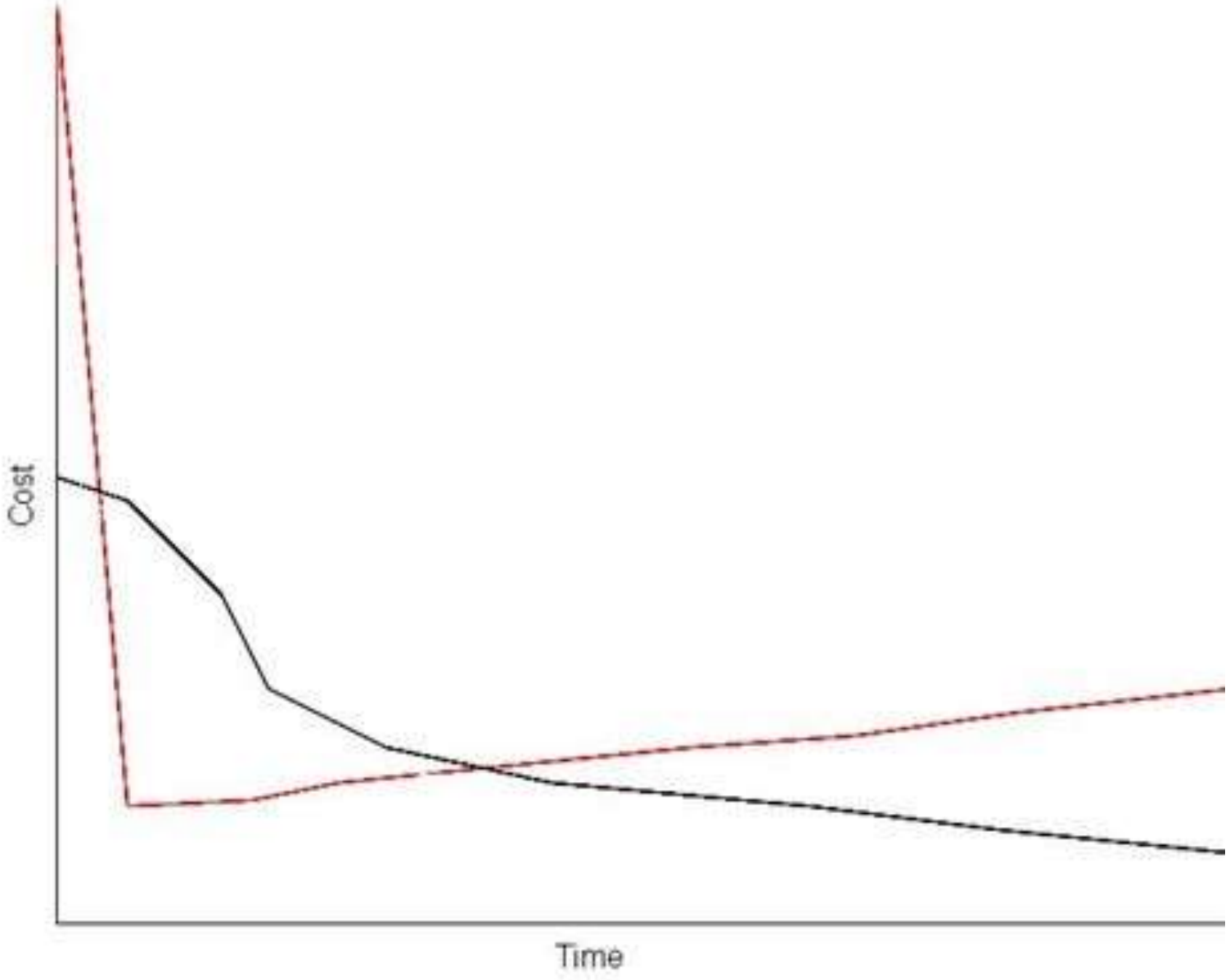
PE – In short

- It is an engineering discipline that deals with all the aspects of developing a large program, by a team of developers
- Proposes a systematic and organized approach to the software development process
- It proposes the use of appropriate techniques and tools taking into consideration:
  - The problem to be solved
  - restrictions
  - available resources

Software vs Hardware costs



Software build (black) vs. buy (red)



Software vs Hardware costs (2)

- Currently the costs for software are higher than the costs of hardware components
- The cost of maintaining a program is usually higher than the cost of creating a program
- Software – represent the programs and the associated documentation

Attributes of a good program

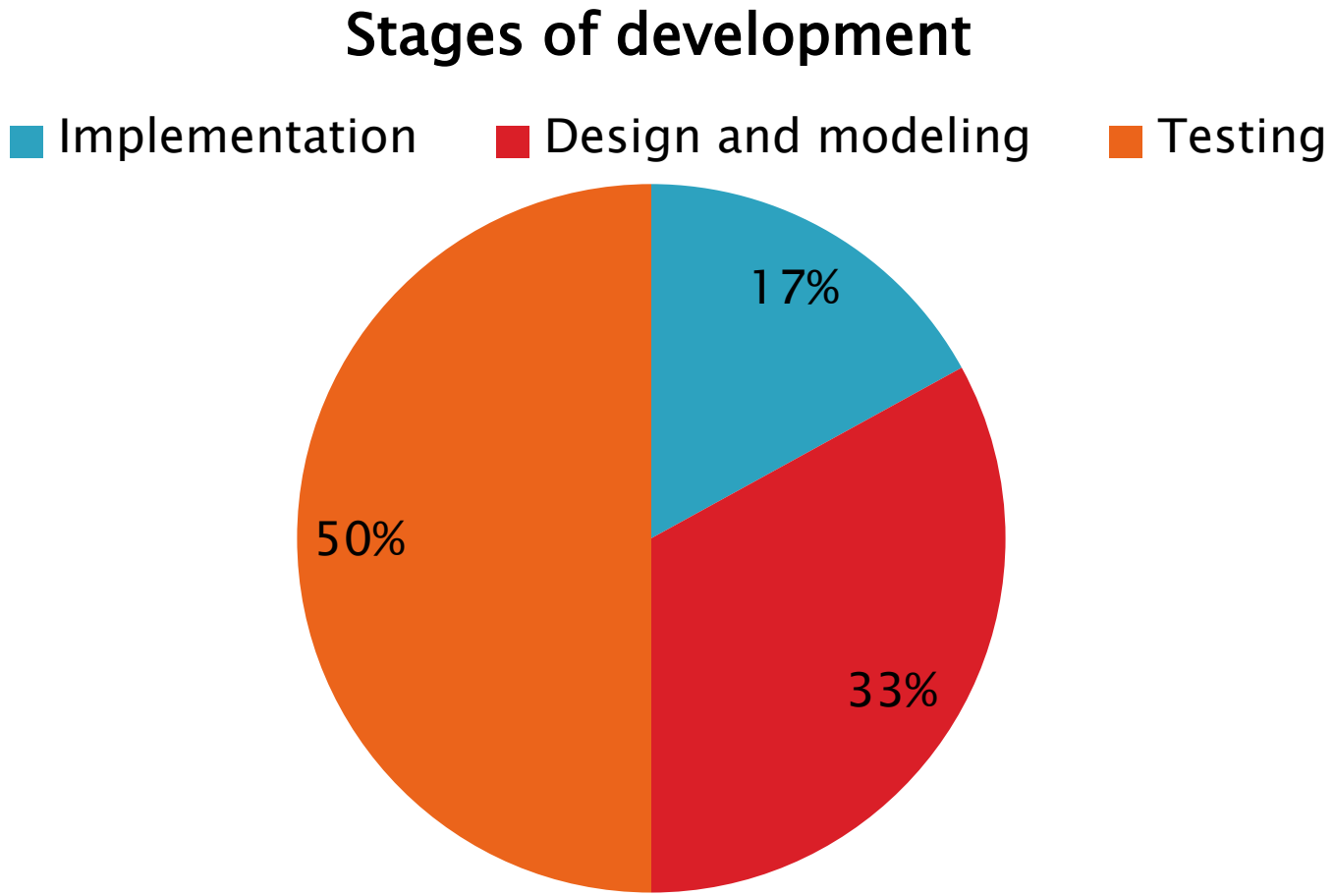
- To provide the requested functionality
- Be easily modified, extended, changed
- Be safe
- Not wasting hardware resources
- Easy to use

Comparison of PE

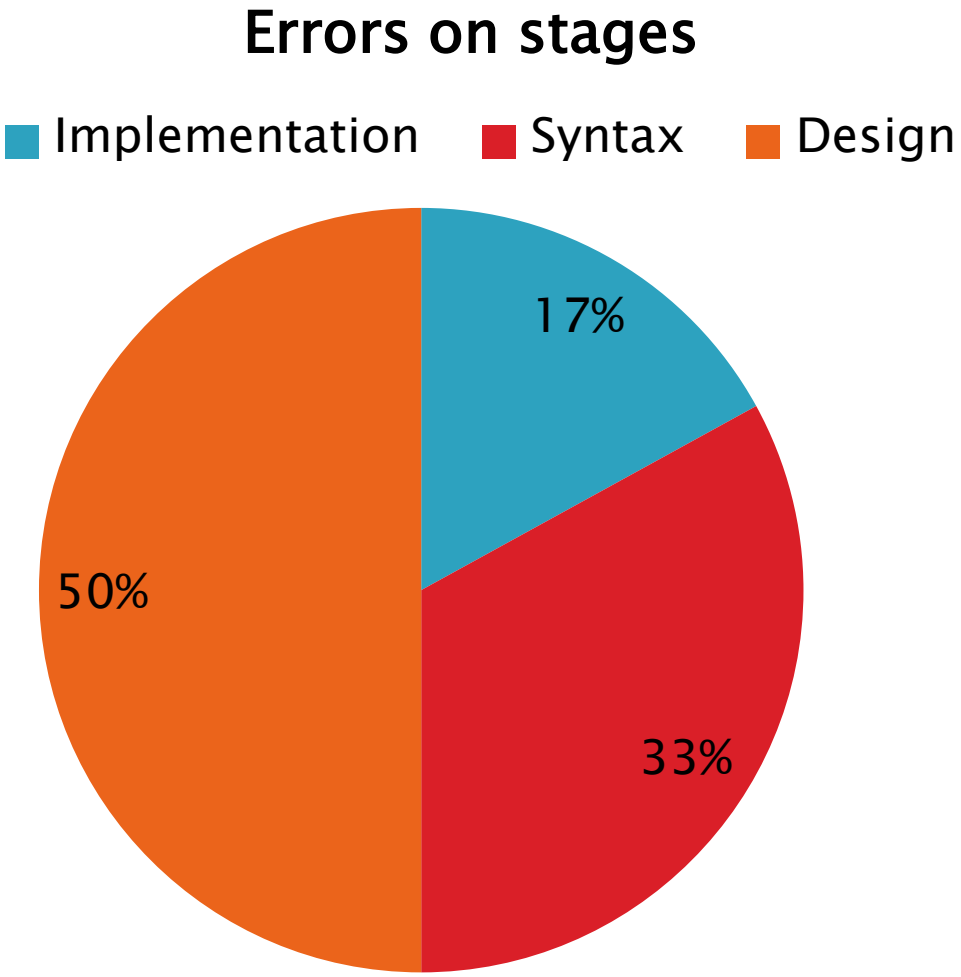
- PE vs computer science
  - Computer science deals with theoretical aspects of software development
  - PE deals with practical aspects of software development

- PE vs systems engineering
  - Systems engineering deals with all aspects of development of computer systems (hardware, software, process engineering)
  - PE is part of systems engineering and is responsible for:
    - Requirements Engineering
    - Architectural and detailed design
    - Implementation
    - Testing
    - Integration
    - Deployment

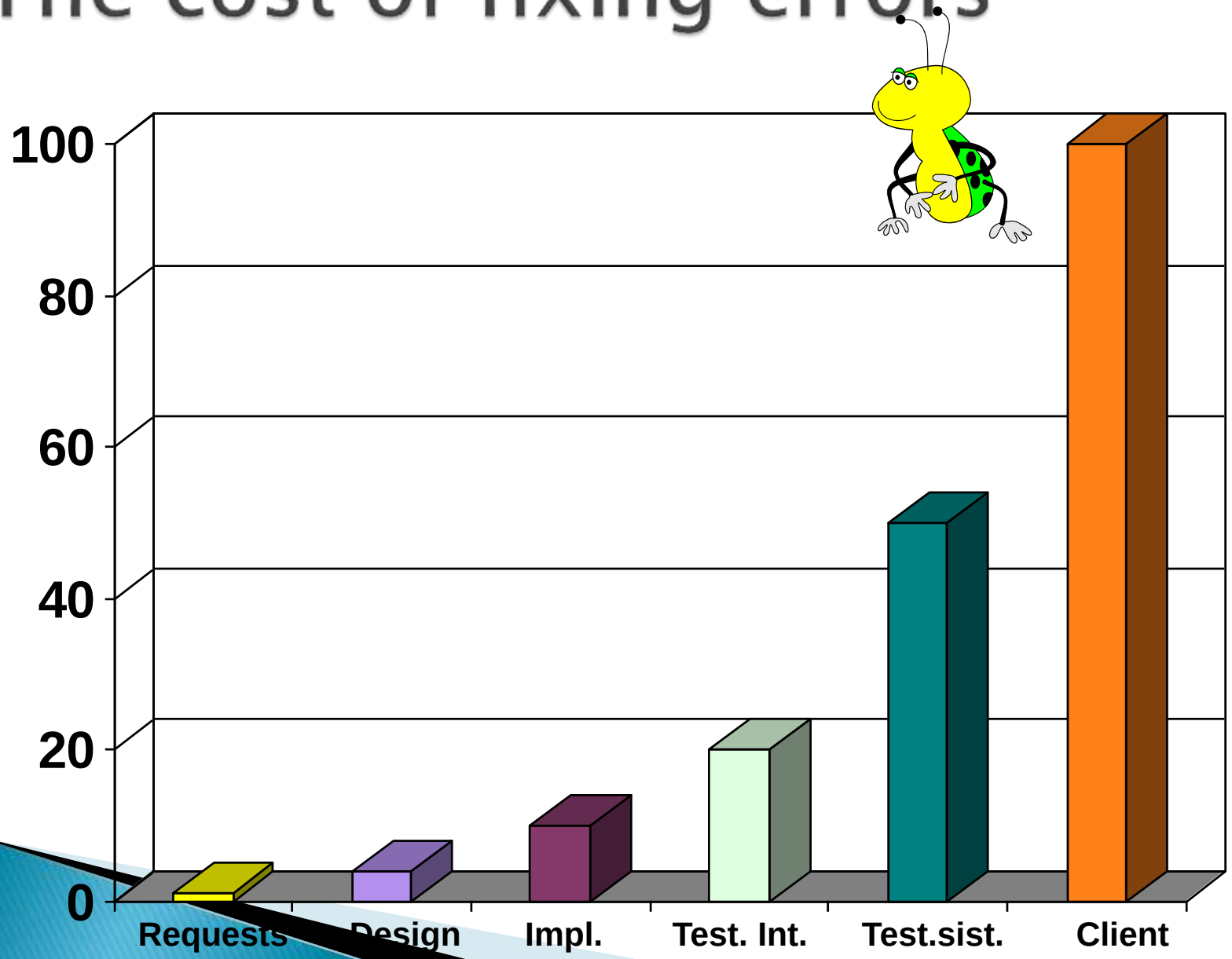
Distribution of costs on stages



Distribution of errors on stages



The cost of fixing errors



Difficulties in PE

- Older systems which need to be maintained and updated
- Heterogeneity of software / hardware systems
- The pressure to deliver software quickly, at a low cost and with good quality

Development Models

- In order to develop a program, the following are necessary:
  - A clear understanding of what is required
  - A set of working methods and tools
  - An action plan

- Action plan template = model development

Software development process

- Requirements Engineering
- Arhitectural design
- Detailed design
- Implementation
- Integration
- Validation
- Verification
- Deployment
- Maintenance

Requirements Engineering

- It sets out what the customer wants the program to do
- The aim is to record the requirements in a clear and more accurate manner
- Problems
  - Communication
  - Negotiation
  - Advise client

Design

- Arhitectural
  - For reasons of complexity, large programs can not be designed and implemented as a single piece
  - The program is divided into modules or simpler components that can be individually addressed
- Detailed
  - Designing each module of the application, in the smallest details

Implementation + integration

- Implementation
  - Detailed design is translated into a programming language
  - This is achieved modularly, on the resulting structure of the architectural design
- Integration
  - Big-bang model
  - Incremental model

Validation and verification

- **Validation**: we ensure that the program meets the user requirements
  - *Are we building the correct product?*

- **Verification**: we ensure that the program is stable and working properly, in terms of development.
  - *Are we building the product correctly?*

Maintenance

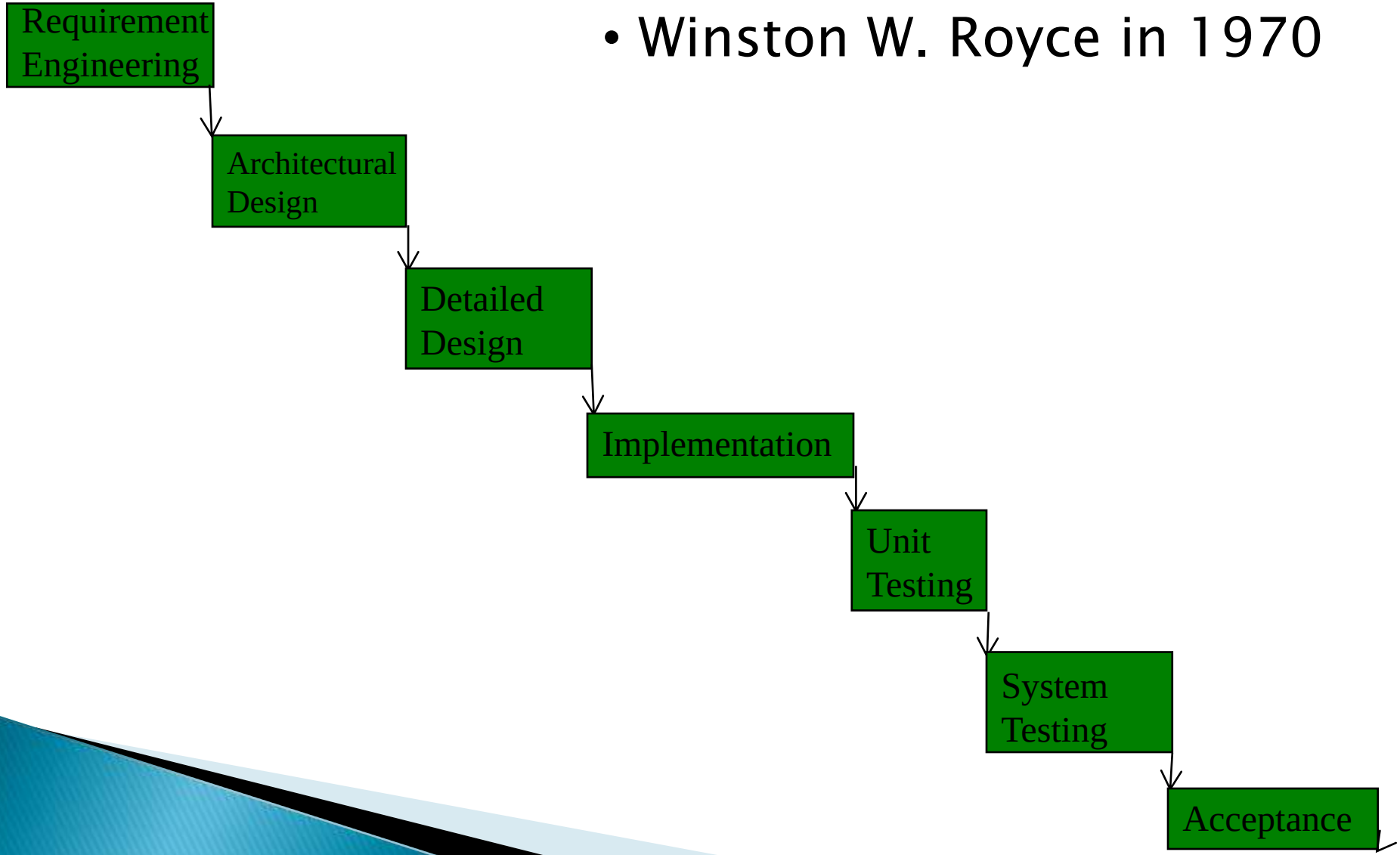
- After delivery
  - Discovered mistakes must be repaired
  - Changes in specifications may appear
  - There may be new requirements
  - The training of those who will use the product

- Maintenance = managing these types of problems

Development models

- *How to perform those activities specified by the stages of software development*
- *Examples of development models:*
  - *Ad hoc: manage as best you can*
  - *Waterfall (with feedback)*
  - *Prototyping*
  - *Spiral*
  - *RUP (Rational Unified Process)*
  - *V-Model*
  - *XP (Extreme Programming)*
  - *Agile, Lean, Scrum*
  - *MDD, AMDD*
  - *TDD*

Waterfall



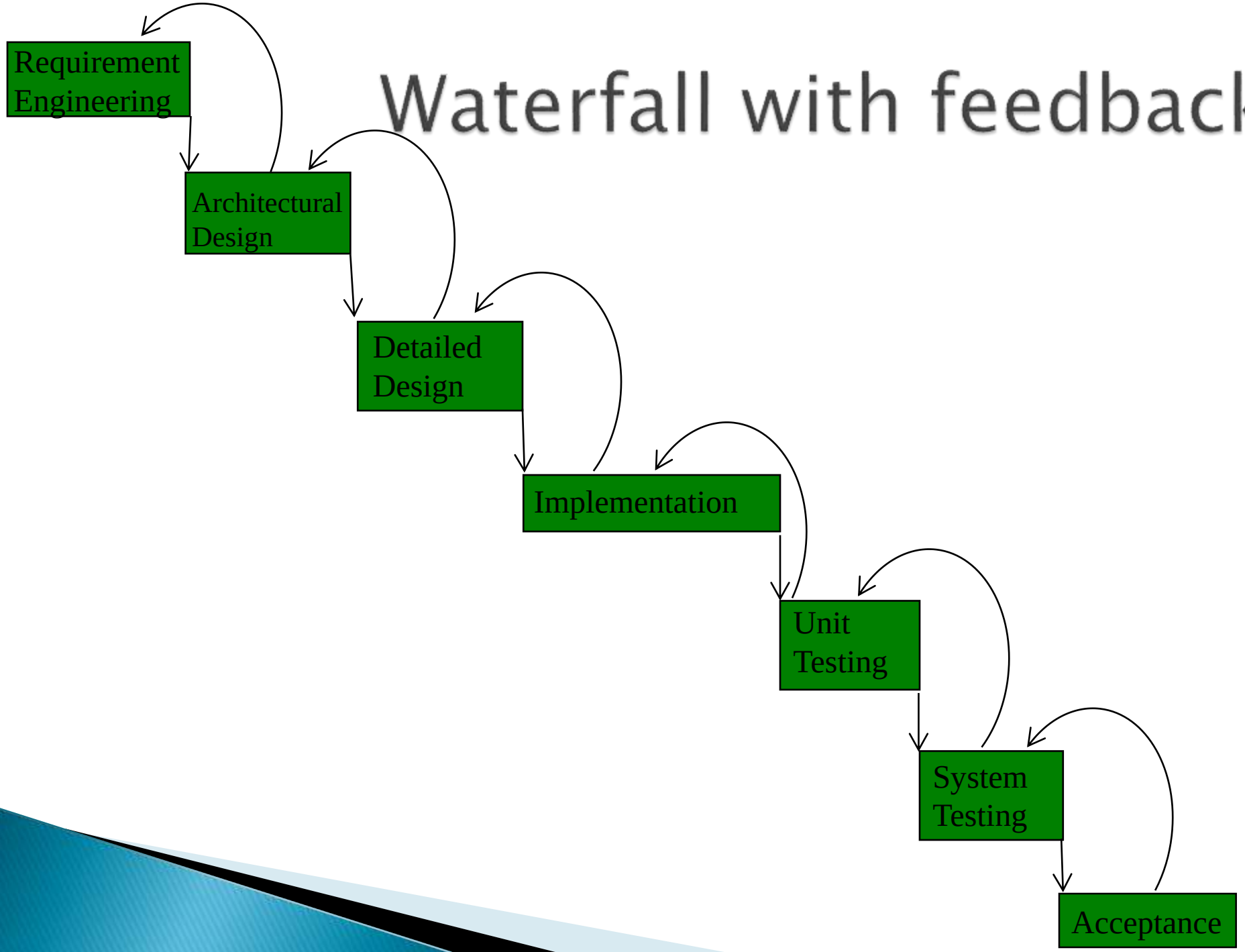
- Winston W. Royce in 1970

Waterfall+-

- +: Split a complex task into smaller steps
- +: Easy to manage and control
- +: Each step results in a well defined product
- +: Always know the phase of the development: what have we finished so far? What else do we have to do further?

- -: Errors are propagated between stages
- -: No repair mechanism for errors

Waterfall with feedback



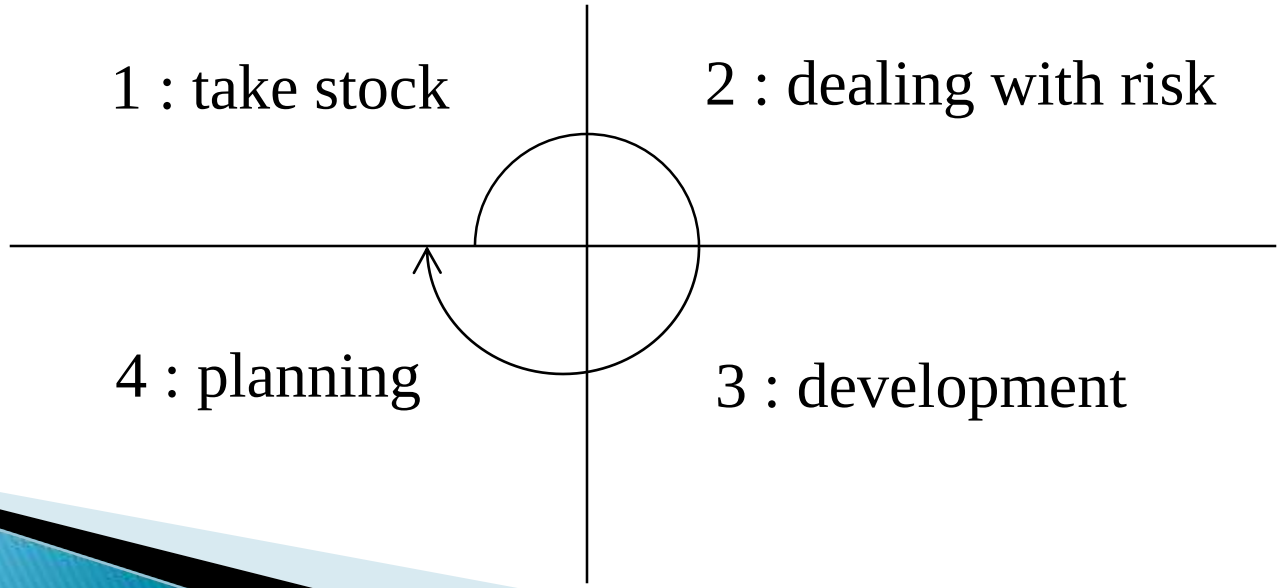
Waterfall with feedback+-

- +: Allows for fixing errors from the previous step
- -: Errors from step i that are discovered to step i + 2 are not remedied
- -: The customer sees the final product only at the end of development

Spiral

- Feasibility study
- Requirements analysis
- Architectural design
- Implementation

At each step, we do the following:



Spiral +-

- +: Keeps benefits from waterfall model
- +: Take into account the notion of risk

- Examples of risks:
  - A competing company launches a similar product
  - An architect leaves team
  - Changing in customer requirements
  - A team does not comply the delivery deadlines

Conclusions

- The statistics prove the importance of using engineering techniques in software development
- PE definitions use keywords such as: *engineering methods, large projects are implemented by teams, safe programs which work effectively, development, maintenance, planning, budgeting*
- Steps necessary for developing large projects
- Development Models: Ad-Hoc, Waterfall, Spiral

Bibliography

- Ovidiu Gheorghieș: http://www.infoiasi.ro/~ogh/files/ip/curs-01.pdf
- Andy Kramek, New Software – Build or Buy? A Personal View: http://weblogs.foxite.com/andykramek /archive/2009/07/25/8674.aspx

Links

- Internet
- Wikipedia
- Failure rate: http://www.it-cortex.com/Stat_Failure_Rate.htm
- RUP in the dialogue with Scrum: http://www.ibm.com/developerworks/rational/library/feb05/krebs/